DATA PREPERATION AND PATTERN DISCOVERY FOR WEB USAGE MINING

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF

Bachelor of Technology
In
Computer Science & Engineering

By
Karan Bhalla & Deepak Prasad

Department of Computer Science & Engineering
National Institute of Technology
Rourkela
2007

DATA PREPERATION AND PATTERN DISCOVERY FOR WEB USAGE MINING

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF

Bachelor of Technology

In

Computer Science & Engineering

By

Karan Bhalla & Deepak Prasad

Under the guidance of:

Professor Dr. S.K. Jena

Department of Computer Science & Engineering

National Institute of Technology

Rourkela

2007

National Institute of Technology

Rourkela

CERTIFICATE

This is to certify that the thesis entitled **"DATA PREPERATION AND PATTERN DISCOVERY FOR WEB USAGE MINING"** submitted by **Mr. Karan Bhalla**, **Roll No: 10306023 and Mr. Deepak Prasad, Roll No: 10306014** in the partial fulfilment of the requirements for the award of **Bachelor of Technology** in **Computer Science and Engineering**, at the National Institute of Technology, Rourkela (Deemed University), is an authentic work carried out by them under my supervision and guidance.

To the best of my knowledge the matter embodied in the thesis has not been submitted to any other university/institute for the award of any degree or Diploma.

Date

Professor Dr. S.K. Jena
Department of Computer Science and Engineering
National Institute of Technology
Rourkela-769008

Acknowledgment

We avail this opportunity to extend our hearty indebtedness to our guide **Professor Dr. S. K. Jena**, Computer Science and Engineering Department, for his valuable guidance, constant encouragement and kind help at different stages for the execution of this dissertation work.

Submitted by:

Karan Bhalla
Roll No: 10306023
Computer Science & Engineering
National Institute of Technology
Rourkela

Deepak Prasad
Roll No: 10306014
Computer Science & Engineering
National Institute of Technology
Rourkela

CONTENTS

Abstract

The World Wide Web (WWW) continues to grow at an astounding rate in both the sheer volume of traffic and the size and complexity of Web sites. The complexity of tasks such as Web site design, Web server design, and of simply navigating through a Web site have increased along with this growth. An important input to these design tasks is the analysis of how a Web site is being used. Usage analysis includes straightforward statistics, such as page access frequency, as well as more sophisticated forms of analysis, such as finding the common traversal paths through a Web site.

Web Usage Mining is the application of data mining techniques to usage logs of large Web data repositories in order to produce results that can be used in the design tasks mentioned above. However, these server logs cannot be used directly for patter discovery and analysis purposes. There are several preprocessing tasks that must be performed prior to applying data mining algorithms to the data collected from server logs. The objective of this paper is to discuss several data preparation techniques in order to identify unique users and user sessions. New heuristics to identify user sessions have been proposed. Also the data mining algorithms that can be applied to this processed data to discover patterns and rules have been discussed. On the basis of implementation of these algorithms, a comparative analysis among some of these algorithms is drawn on a 2-dimensional graph.

Chapter 1

INTRODUCTION

BACKGROUND

OBJECTIVE

BACKGROUND

The World Wide Web (WWW) continues to grow at an astounding rate in both the sheer volume of traffic and the size and complexity of Web sites. The complexities of tasks such as Web site design, Web server design, and of simply navigating through a Web site have increased along with this growth. An important input to these design tasks is analysis of how a Web site is being used. Usage analysis includes straightforward statistics, such as page access frequency, as well as more sophisticated forms of analysis, such as finding the common traversal paths through a Web site. Usage information can be used to restructure a Web site in order to better serve the needs of users of a site. Long convoluted traversal paths or low usage of a page with important site information could suggest that the site links and information are not laid out in an intuitive manner. The design of a physical data layout or caching scheme for a distributed or parallel Web server can be enhanced by knowledge of how users typically navigate through the site. Usage information can also be used to directly aide site navigation by providing a list of "popular" destinations from a particular Web page.

Web Usage Mining is the application of data mining techniques to large Web data repositories in order to produce results that can be used in the design tasks mentioned above. Some of the data mining algorithms that are commonly used in Web Usage Mining are association rule generation, sequential pattern generation, and clustering. *Association Rule mining* techniques discover unordered correlations between items found in a database of transactions. In the context of Web Usage Mining a transaction is a group of Web page accesses, with an item being a single page access.

The problem of discovering *sequential patterns* is that of finding inter-transaction patterns such that the presence o f a set of items is followed by another item in the time-stamp ordered transaction set. By analyzing this information, a Web Usage Mining system can determine temporal relationships among data items.

Clustering analysis allows one to group together users or data items that have similar characteristics. Clustering of user information or data from Web server logs can facilitate the development and execution of future marketing strategies, both online and off-line, such as automated return mail to visitors falling within a certain cluster, or dynamically changing a particular site for a visitor on a return visit, based on past classification oft hat visitor.

Mining for knowledge from Web log data has the potential of revealing information of great value. While this certainly is an application of existing data mining algorithms, e.g. discovery of association rules or sequential patterns, the overall task is not one of simply adapting

existing algorithms to new data. Ideally, the input for the Web Usage Mining process is a file, referred to as a *user session file* in this project, that gives an exact accounting of who accessed the Web site, what pages were requested and in what order, and how long each page was viewed. A user session is considered to be all of the page accesses that occur during a single visit to a Web site. The information contained in a raw Web server log does not reliably represent a user session file for a number of reasons that will be discussed in this paper. Specifically, there are a number of difficulties involved in cleaning the raw server logs to eliminate outliers and irrelevant items, reliably identifying unique users and user sessions within a server log, and identifying semantically meaningful transactions within a user session.

OBJECTIVE

This project presents different Data Mining techniques and algorithms used in mining information from Web server logs or in the process of Web Usage data mining. The specific contribution in this regard includes implementation of some of these algorithms, and a comparative analysis between some of these algorithms using graphs.

In addition to this, the paper also discusses the several data preparation techniques and algorithms that can be used to convert raw Web server logs into user session files in order to perform Web Usage Mining. The specific contributions include, (i) development of models to encode both the Web site developer's and users' view of how a Web site should be used, (ii) discussion of heuristics that can be used to identify Web site users, user sessions, and page accesses that are missing from a Web server log,

Chapter 2

DATA MINING

Data mining, *the extraction of hidden predictive information from large databases*, is a powerful new technology with great potential to help companies focus on the most important information in their data warehouses [3]. Data mining tools predict future trends and behaviours, allowing businesses to make proactive, knowledge-driven decisions. The automated, prospective analyses offered by data mining move beyond the analyses of past events provided by retrospective tools typical of decision support systems. Data mining tools can answer business questions that traditionally were too time consuming to resolve. They scour databases for hidden patterns, finding predictive information that experts may miss because it lies outside their expectations [3].

Most companies already collect and refine massive quantities of data. Data mining techniques can be implemented rapidly on existing software and hardware platforms to enhance the value of existing information resources, and can be integrated with new products and systems as they are brought on-line. When implemented on high performance client/server or parallel processing computers, data mining tools can analyze massive databases to deliver answers to questions such as, "Which clients are most likely to respond to my next promotional mailing, and why?"

2.1. THE DATA MINING PROCESS

The data mining process [3] is much likely to include the following steps:

1) *Requirement analysis:* The enterprise decision makers need to formulate goals that the data mining process is expected to achieve. The business problem must be clearly defined. One cannot use data mining without a good idea of what kind of outcomes the enterprise is looking for, since the technique to be used and the data that is required are likely to be different for different goals. Furthermore, if the objectives have been clearly defined, it is easier to evaluate the results of the project. Once the goals have been agreed upon, the following further steps are needed.
2) *Data selection and collection:* this step may include finding the best source databases for the data that is required. If the enterprise has implemented a data warehouse, then most of the data could be available there. If the data is not available in the warehouse or the enterprise does not have a warehouse, the source OLTP systems need to be identified and the required information extracted and stored in some temporary systems.

3) *Cleaning and preparing data:* This may not be an onerous task if a data warehouse containing the required data exists, since most of this must have already been done when data was loaded in the warehouse. Otherwise, this task can be very resource intensive and sometimes more than 50% of effort in a data mining project is spent on this step. Essentially, a data store that integrated data from a number of databases may need to be created. When integrating data, one often encounters problems like identifying data, dealing with missing data, data conflicts and ambiguity.
4) *Data mining exploration and validation:* Once appropriate data has been collected and cleaned, it is possible to start data mining exploration. Assuming that the user has access to one or more data mining tools, a data mining model may be constructed based on the enterprise's needs. It may be possible to take a sample of data and apply a number of relevant techniques. For each technique the result should be evaluated and their significance interpreted. This is likely to be an iterative process which should lead to selection of one or more techniques that are suitable for further exploration.
5) *Implementation, evaluating, and monitoring:* Once a model has been selected and validated, the model can be implemented for use by the decision makers. This may involve software development for generating reports, or for results visualization and explanation, for managers. It may be that more than one technique is available for the given data mining task. It is then important to evaluate the results and choose the best technique Furthermore, there is a need for regular monitoring of the performance of the techniques that have been implemented.
6) *Result visualization:* Explaining the result of data mining to the decision makers is important step in data mining process. Most commercial data mining tools include data visualization modules. These tools are often vital in communication the data mining result to the managers; although a problem dealing with a number of dimensions must be visualized using a 2D computer screen or printout.

2.2. DATA MINING TECHNIQUES

Data mining employs a number of techniques[3] including the following:

- Association Rule mining
- Supervised classification
- Cluster analysis

- Sequential pattern discovery
- Regression

Only the first three from above are employed in Web Usage mining and will be studied in detail subsequently.

2.2.1. Association Rule Mining:

Association approaches address a class of problems typified by a market-basket analysis. Classic market-basket analysis treats the purchase of a number of items (for example, the contents of a shopping basket) as a single transaction. The goal is to find trends across large numbers of transactions that can be used to understand and exploit natural buying patterns. This information can be used to adjust inventories, modify floor or shelf layouts, or introduce targeted promotional activities to increase overall sales or move specific products. While these approaches had their origins in the retail industry, they can be applied equally well to services that develop targeted marketing campaigns or determine common (or uncommon) practices. In the financial sector, association approaches can be used to analyze customers' account portfolios and identify sets of financial services that people often purchase together. They may be used, for example, to create a service "bundle" as part of a promotional sales campaign.

Associations are written as A |=> B, where A is called the antecedent or left-hand side (LHS), and B is called the consequent or right-hand side (RHS). For example, in the association rule "If people buy a hammer then they buy nails," the antecedent is "buy a hammer" and the consequent is "buy nails." It's easy to determine the proportion of transactions that contain a particular item or item set: simply count them. The frequency with which a particular association (e.g., the item set "hammers and nails") appears in the database is called its *support* or *prevalence.* If, say, 15 transactions out of 1,000 consist of "hammer and nails," the support for this association would be 1.5%. A low level of support (say, one transaction out of a million) may indicate that the particular association isn't very important or it may indicate the presence of bad data (e.g., "male and pregnant").

Support: Support for X is the number of times it appears in the database divided by N and support for X and Y together is the number of times they appear together in the database divided by N.

Support(X) = (Number of times X appears)/N = P(X)

Support (XY) = (Number of times X and Y appears together)/N = P(X ∩ Y)

Confidence: Confidence for X→Y is defined as the ratio of the support for X and Y together to the support for X.

Confidence of (X→Y) = Support (XY)/Support(X) = P(X ∩ Y)/P(X) = P (Y|X)

2.2.1.1 The Apriori Algorithm

Input:

transa.txt

TID	**A**	**B**	**C**	**D**	**E**	**F**	**G**	**H**
T1	0	1	0	1	1	0	1	0
T2	1	1	0	1	0	1	0	0
T3	0	1	1	0	0	1	0	1
T4	1	0	0	0	1	0	0	0
T5	0	1	0	1	0	0	1	1
T6	1	1	0	0	1	0	1	0

The above representation of the input is known as the vertical representation.

config.txt

Number of items: 5
Number of transactions: 5
Minimum support: 40
Line ignored by apriori algorithm

Output:

Input configuration: 5 items, 5 transactions, minsup = 40%

Frequent 1-itemsets:
[1, 2, 3, 4, 5]
Frequent 2-itemsets:
[1 2, 1 3, 1 4, 1 5, 2 3, 2 4, 3 4, 3 5, 4 5]
Frequent 3-itemsets:
[1 2 3, 1 2 4, 1 3 4, 1 3 5, 1 4 5, 2 3 4, 3 4 5]
Frequent 4-itemsets:
[1 2 3 4, 1 3 4 5]

Execution time is: 40 milliseconds.

Algorithm:

The Apriori algorithm finds the frequent sets L In Database D.

- Find frequent set L_{k-1}.
- Join Step.
  - C_k is generated by joining $L_{k-1}$with itself
- Prune Step.
  - Any $(k-1)$ -itemset that is not frequent cannot be a subset of a frequent k -itemset, hence should be removed.

where

- (C_k: Candidate itemset of size k)
- (L_k: frequent itemset of size k)

Apriori Pseudocode

Apriori (T,e)

$L_1 \leftarrow$ *{large 1-itemset}*

$k \leftarrow 2$

while $L_{k-1} \neq \Phi$

$C_k \leftarrow$ Generate (L_{k-1})

For transactions t $\in$ T

$C_t \leftarrow$ Subset(C_kf)

For candidate c $\in$ C_t

count[c] $\leftarrow$ count[c] + 1

$L_k \leftarrow \{c \in C_k \mid count[c] \geq e\}$

k $\leftarrow$ k+1

return $\bigcup_k L_k$

2.2.1.2 Dynamic Itemset Counting (DIC) Algorithm

It is an alternate to Apriori Itemset generation in which Itemsets are dynamically added or deleted as transactions are read. It relies on the fact that for an itemset to be frequent, all of its subsets must also be frequent, so we only examine those itemsets whose subsets are all frequent.

Algorithm stops after every *M* transactions to add more itemsets.

Train analogy: There are stations every *M* transactions. The passengers are itemsets. Itemsets can get on at any stop as long as they get off at the same stop in the next pass around the database. Only itemsets on the train are counted when they occur in transactions. At the very beginning we can start counting 1-itemsets, at the first station we can start counting some of the 2-itemsets. At the second station we can start counting 3-itemsets as well as any more 2-itemsets that can be counted and so on.

Input:

transa.txt: same as apriori algorithm above

config.txt

Number of items: 5
Number of transactions: 5

Minimum support: 40

Size of step M: 5

Output:

Input configuration: 5 items, 5 transactions, minsup = 40%

Processing step M number: 1..2..3..4..5..

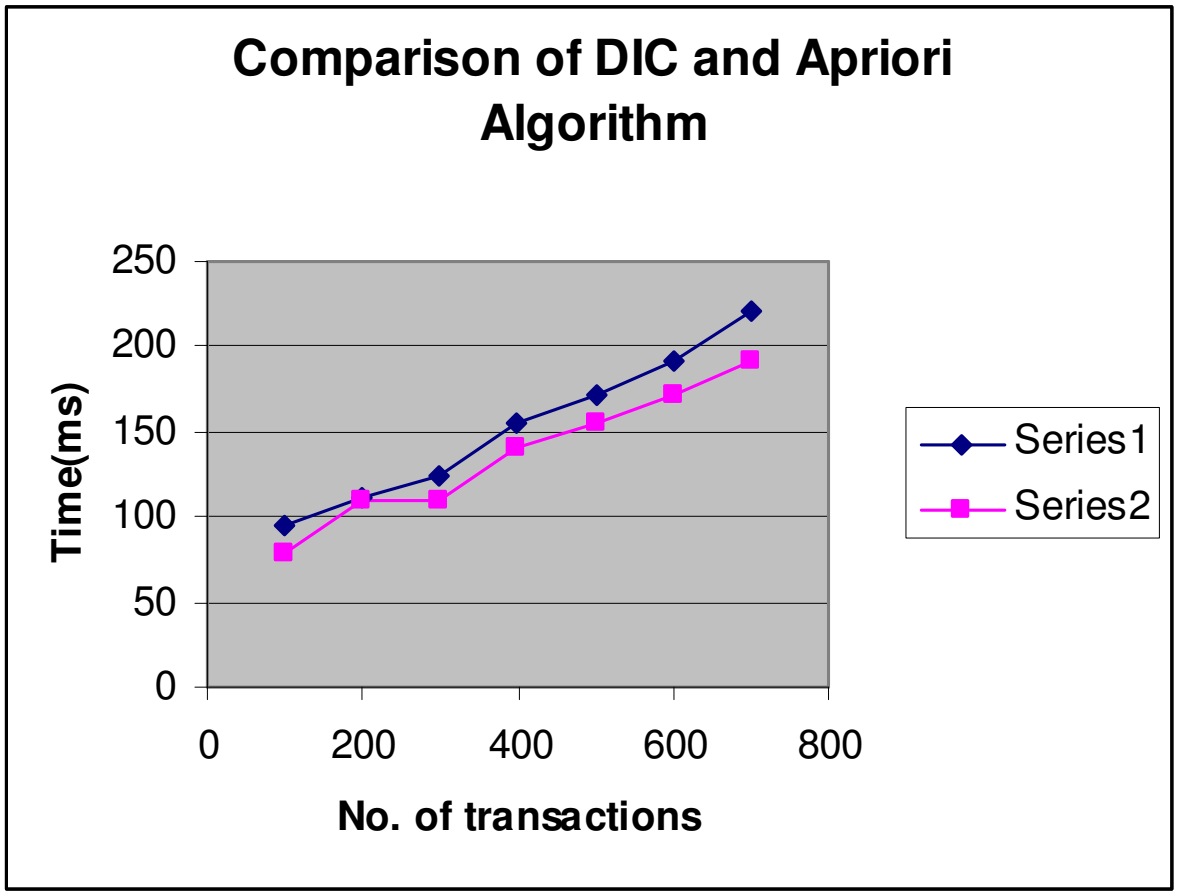
Frequent 1-itemsets:
[1, 2, 3, 4, 5]
Frequent 2-itemsets:
[1 2, 1 3, 1 4, 1 5, 2 3, 2 4, 3 4, 3 5, 4 5]
Frequent 3-itemsets:
[1 2 3, 1 2 4, 1 3 4, 1 3 5, 1 4 5, 2 3 4, 3 4 5]
Frequent 4-itemsets:
[1 2 3 4, 1 3 4 5]

Execution time is: 35 milliseconds.

Algorithm:

Algorithm:

1. Mark the empty itemset with a solid square. Mark all the 1-itemsets with dashed circles. Leave all other itemsets unmarked.
2. While any dashed itemsets remain:
    1. Read *M* transactions (if we reach the end of the transaction file, continue from the beginning). For each transaction, increment the respective counters for the itemsets that appear in the transaction and are marked with dashes.
    2. If a dashed circle's count exceeds *minsup*, turn it into a dashed square. If any immediate superset of it has all of its subsets as solid or dashed squares, add a new counter for it and make it a dashed circle.
    3. Once a dashed itemset has been counted through all the transactions, make it solid and stop counting it.



Following are the advantages of DIC over Apriori algorithm:

1) *Reduce the no. of comparisons:* The DIC algorithm reduces the number of scans required by not just doing one scan for the frequent 1-itemset and another for the frequent 2-itemset but combining the counting for a number of itemsets as soon as it appears that it might be necessary to count it.
2) DIC algorithm works well when the data is relatively homogeneous throughout the file since it starts the 2-itemset count before having a final 1-itemset count. In case of non-homogeneous data distribution, the algorithm may not identify an itemset to be large until most of the database has been scanned. It is possible to randomize the transaction data in such cases, although this is not always possible.
3) Essentially DIC attempts to finish itemset counting in two scans of the database while Apriori would often take three or more scans.

2.2.2 Classification Techniques

It is an important data mining technique that has its origins in machine learning. Supervised classification is appropriate to use if the data is known to have small number of classes, the classes are already known and some training data with their classes known is available. The model based on the training data may then be used to assign a new object to a predefined class.

Supervised classification can be used in predicting the class to which an object or individual is likely to belong. This is useful, for example, in predicting whether an individual is likely to respond to a direct mail solicitation, in identifying a good candidate for a surgical procedure, or in identifying a good risk for granting a loan or insurance. One of the most widely used supervised classification technique is the decision tree. It is widely used because it generates easily understandable rules for classifying data.

Classification processes in which classes have been pre-defined need a method that will train the classification system to allocate objects to the classes. The training is based on training sample, a set of sample data where for each sample the class is already known. We assume each object to have a number of attributes, one of which tells us which class the object belongs to. This attribute is known for the training data but data other than the training data we assume that the value of the attribute is unknown and is to be determined by the classification method. This attribute may be considered as the output of all other attributes and is often referred to as the output attribute or the dependent attribute. The attribute other than the output attribute are called the input attributes or the independent attributes.

Attributes whose domain is numerical is called *numerical* attributes while attributed whose domain is not numerical are called *categorical* attributes.

Two algorithms are extensively used in field of Classification:

- Decision Tree
- Naïve Bayes method of classification

2.2.3 Cluster Analysis

We like to organize observations or objects or things into meaningful groups so that we are able to make comments about the groups rather than individual objects. Such groupings are often rather convenient since we can talk about a small number of groups rather than a large number of objects although certain details are necessarily lost because objects in each group are not identical.

Classification and Clustering are important techniques that partition objects that have many attributed into meaningful disjoint subgroups so that objects in each group are most similar to each other in the values of their attributes than they are to object in other groups. A major difference in case of both is that in case of supervised classification, the classes are pre-defined, the user already knows what classes there are, and some training data that is already labeled by their class membership is available to train or build a model. The classification problem then is to build a model that would be able to label or classify newly encountered data.

Computing distance

Most cluster analysis methods are based on measuring similarity between objects by computing the distance between each pair. Let the distance between two points *x* and *y* be D(x, y). Following are a number of distance measures:

Euclidian distance

The largest valued attribute may dominate the distance. It is therefore essential that the attributes are properly scaled.

$$D(x, y) = (\sum (x_i - y_i)^2)^{1/2}$$

It is possible to use this distance measure without the square root if one wanted to place greater weight on difference that are large.

Manhattan distance

The largest valued attribute can dominate the distance, although not as much as in case of Euclidian distance.

$$D(x, y) = \sum |x_i - y_i|$$

2.2.3.1 The K-Means Clustering method

It is the simplest and most popular classical clustering method. The method is called the K-Means since each of the K clusters is represented by the mean of the objects (called the centroid) within it. It is also called the *centroid method* since at each step the centroid point of each cluster whose centroid is assumed to be known and each of the remaining points are allocated to the cluster whose centroid is closest to it. Once this allocation is completed, the centroids of the clusters are recomputed using simple means and the process of allocating point to each cluster is repeated until there is no change in the clusters.

The K-Means method uses the *Euclidian distance* measure, which appears to work well with compact clusters.

Input :

Number of clusters, and the 13 X 4 array given below. Each row in the array is an object.

0.10 0.00 9.60 5.60

1.40 1.30 0.00 3.80

1.20 2.50 0.00 4.80

2.30 1.50 9.20 4.30

1.70 0.70 9.60 3.40

0.00 3.90 9.80 5.10

6.70 3.90 5.50 4.80

0.00 6.30 5.70 4.30

5.70 6.90 5.60 4.30

0.00 2.20 5.40 0.00

3.80 3.50 5.50 9.60

0.00 2.30 3.60 8.50

4.10 4.50 5.80 7.60

Output

Points' cluster IDs are: 0 1 1 2 2 0 3 4 4 2 3 1 3

cluster 0: (0.050000 1.950000 9.700001 5.350000)

cluster 1: (0.866667 2.033333 1.200000 5.700000)

cluster 2: (1.333333 1.466667 8.066667 2.566667)

cluster 3: (4.866667 3.966666 5.600000 7.333333)

cluster 4: (2.850000 6.600000 5.650000 4.300000)

The different objects are classified and presented as output.

Algorithm

1. Select the number of clusters. Let this number be *k.*
2. Pick *k* seeds as centroids of the *k* clusters, The seeds may be picked randomly unless the user has some insight into the data.
3. Compute the Euclidian distance of each object in the dataset from each of the centroids.
4. Allocate each object to the cluster it is nearest to based on the distances computed in the previous step.
5. Compute the centroids of the clusters by computing the means of the attribute values of the objects in each cluster.
6. Check if the stopping criterion has the met. If yes, go to Step 7. If not, go to step 3.

Chapter 3

WEB MINING

With the explosive growth of information sources available on the World Wide Web, it has become increasingly necessary for users to utilize automated tools to find the desired information resources, and to track and analyze their usage patterns [6]. These factors give rise to the necessity of creating server side and client side intelligent systems that can effectively mine for knowledge. Web mining can be broadly defined as the discovery and analysis of useful information from the World Wide Web. This describes the automatic search of information resources available on line, i.e. *Web content mining [6]*, and the discovery of user access patterns from Web servers, i.e. *Web usage mining [6]*.

3.1 WEB CONTENT MINING

The lack of structure that permeates the information sources on the World Wide Web makes automated discovery of Web based information difficult [6]. Traditional search engines such as Lycos, Alta Vista, WebCrawler, ALIWEB, MetaCrawler, and others provide some comfort to users but do not generally provide structural information nor categorize filter or interpret documents. A recent study provides a comprehensive and statistically thorough comparative evaluation of the most popular search engines.

3.1.1 Agent-Based Approach

Generally agent-based Web mining systems can be placed into the following three categories:

Intelligent Search Agents: Several intelligent Web agents have been developed that search for relevant information using domain characteristics and user profiles to organize and interpret the discovered information.

Information Filtering/Categorization: A number of Web agents use various information retrieval techniques and characteristics of open hypertext Web documents to automatically retrieve later and categorize.

Personalized Web Agents: This category of Web agents learn user preferences and discover Web information sources based on these preferences, and those of other individuals with similar interest.

3.1.2 Database Approach

Database approaches to Web mining have focused on techniques for organizing the semi-structured data on the Web into more structured collections of resources, and using standard database querying mechanisms and data mining techniques to analyze it:

Multilevel Databases: The main idea behind this approach is that the lowest level of the database contains semi-structured information stored in various Web repositories, such as hypertext documents. At the higher level(s) metadata or generalizations are extracted from lower levels and organized in structured collections, i.e. relational or object-oriented Databases.

Web Query Systems: Many Web based query systems and languages utilize standard database query languages such as SQL, structural information about Web documents, and even natural language processing for the queries that are used in World Wide Web searches.

3.2 WEB USAGE MINING

Web usage mining is the automatic discovery of user access patterns from Web servers. Organizations collect large volumes of data in their daily operations, generated automatically by Web servers and collected in server access logs. Other sources of user information include referrer logs which contain information about the referring pages for each page reference, and user registration or survey data gathered via CGI scripts [6]. Analyzing such data can help organizations determine the life time value of customers, cross marketing strategies across products, and effectiveness of promotional campaigns, among other things. It can also provide information on how to restructure a Web site to create a more effective organizational presence and shed light on more effective management of workgroup communication and organizational infrastructure. For selling advertisements on the World Wide Web analyzing user access patterns helps in targeting ads to specific groups of users. Most existing Web analysis tools provide mechanisms for reporting user activity in the servers and various forms of data filtering. Using such tools it is possible to determine the number of accesses to the server and to individual files the times of visits and the domain names and URLs of users. However these tools are designed to handle low to moderate traffic servers, and usually provide little or no analysis of data relationships among the accessed files and directories within the Web space. More sophisticated systems and techniques for discovery and analysis of patterns are now emerging. These tools can be placed into two main categories as discussed below:

Pattern Discovery Tool: The emerging tools for user pattern discovery use sophisticated techniques from AI, data mining, psychology, and information theory, to mine for knowledge from collected data.

Pattern Analysis Tools: Once access patterns have been discovered analysts need the appropriate tools and techniques to understand, visualize, and interpret these patterns.

Chapter 4

DATA PREPROCESING MODEL FOR WEB USAGE MINING

Web Usage Mining process are an encoding of the site developer's view of browsing behavior and an encoding of the actual browsing behaviors. These inputs are derived from the site files and the server logs respectively [1].

4.1 Developer's Model

The Web site developer's view of how the site should be used is inherent in the structure of the site. Each link between pages exists because the developer believes that the pages are related in some way. Also, the content of the pages themselves provide information about how the developer expects the site to be used. Hence, an integral step of the preprocessing phase is the classifying of the site pages and extracting the site topology from the HTML files that make up the web site. The topology of a Web site can be easily obtained by means of a site "crawler" that parses the HTML files to create a list of all of the hypertext links on a given page, and then follows each link until all of the site pages are mapped. The pages are classified as follows:

- Head Page - a page whose purpose is to be the first page that users visit, i.e. "home" pages.
- Content Page - a page that contains a portion oft he information content that the Web site is providing.
- Navigation Page - a page whose purpose is to provide links to guide users on to content pages.
- Look-up Page - a page used to provide a definition or acronym expansion.
- Personal Page - a page used to present information of a biographical or personal nature for individuals associated with the organization running the Web site.

4.2 Users' Model

Analogous to each of the common physical characteristics for the different page types, there is expected to be common usage characteristics among different users. The reference length of a page is the amount of time a user spends viewing that particular page for a specific log entry.

In order to group individual Web page references into meaningful transactions for the discovery of patterns such as association rules, an underlying model of the user's browsing behavior is needed. For the purposes of association rule discovery, it is really the content page references

that are of interest. The other page types are just to facilitate the browsing of a user while searching for information, and will be referred to as auxiliary pages. What is merely an *auxiliary* page for one user may be a *content* page for another. Transaction identification assumes that user sessions have already been identified. Using the concept of auxiliary and content page references, there are two ways to define transactions, as shown in Fig. 3. The first would be to define a transaction as all of the auxiliary references up to and including each content reference for a given user. Mining these *auxiliary-content* transactions would essentially give the common traversal paths through the web site to a given content page. The second method would be to define a transaction as all of the content references for a given user. Mining these *content-only* transactions would give associations between the content pages of a site, without any information as to the path taken between the pages. It is important to note that results generated from content-only transactions only apply when those pages are used as content references. For example, an association rule, A → B, is normally taken to mean that when A is in a set, so is B. However, if this rule has been generated with content-only transactions, it has a more specific meaning, namely that A implies B only when both A and B are used as content references. This property allows data mining on content-only transactions to produce rules that might be missed by including all oft he page references in a log. If users that treat page A as an auxiliary page do not generally go on to page B, inclusion of the auxiliary references into the data mining process would reduce the confidence of the rule A → B, possibly to the point where it is not reported. Depending on the goals of the analysis, this can be seen as an advantage or a disadvantage.

4.3 Preprocessing

The preprocessing model shown in Fig 4.1 is as proposed in the paper [1]. In subsequent sections we will propose certain heuristics in addition to the ones already proposed in [1] for identifying user sessions. Our study only concerns till forming the *user session files*. We in this paper does not discuss about transaction Identification.

The inputs to the preprocessing phase are the server logs, site files, and optionally usage statistics from a previous analysis. The outputs are the user session file, transaction file, site topology, and page classifications. One of the major impediments to creating a reliable user session file is browser and proxy server caching. Current methods to collect information about cached references include the use of cookies and cache busting. Cache busting is the practice of preventing browsers from using stored local versions of a page, forcing a new download of a page from the server every time it is viewed. None of these methods are without serious

drawbacks. Cookies can be deleted by the user and cache busting defeats the speed advantage that caching was created to provide, and is likely to be disabled by the user. Another method to identify users is user registration. Registration has the advantage of being able to collect additional demographic information beyond what is automatically collected in the server log, as well as simplifying the identification of user sessions. However, again due to privacy concerns, many users choose not to browse sites that require registration and logins, or provide false information.

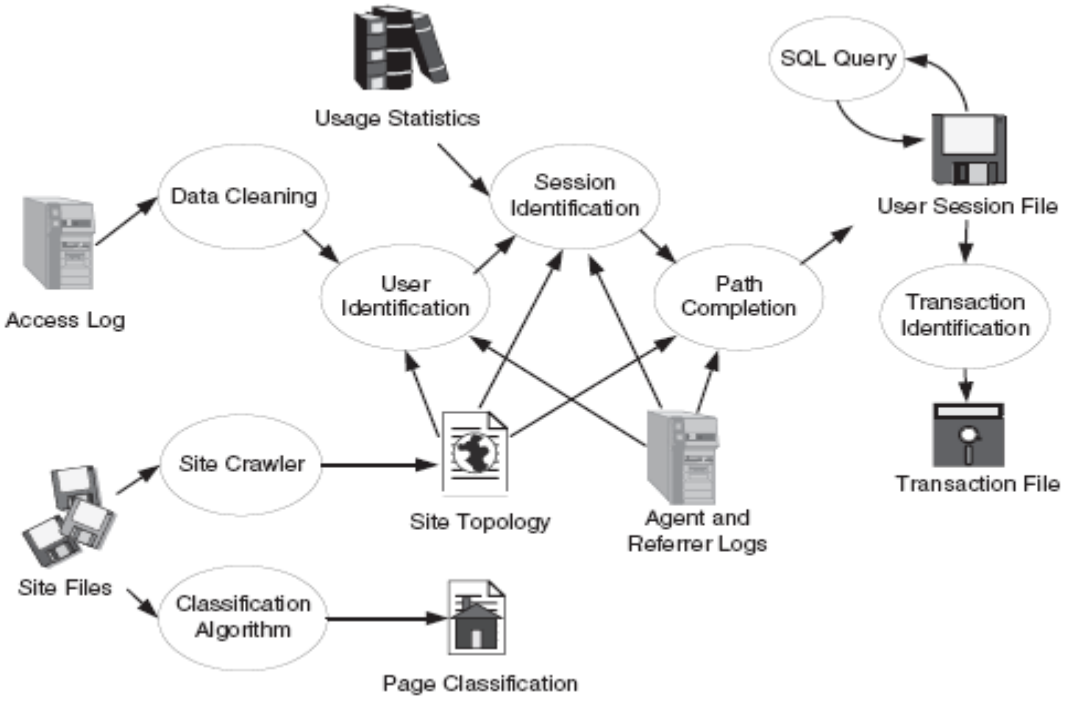


Figure 4.1

4.3.1 Data Cleaning

Techniques to clean a server log to eliminate irrelevant items are of importance for any type of Web log analysis, not just data mining. The discovered associations or reported statistics are only useful if the data represented in the server log gives an accurate picture of the user accesses to the Web site. The HTTP protocol requires a separate connection for every file that is requested from the Web server. Therefore, a user's request to view a particular page often results in several log entries since graphics and scripts are down-loaded in addition to the HTML file. In most cases,

only the log entry oft he HTML file request is relevant and should be kept for the user session file. This is because, in general, a user does not explicitly request all of the graphics that are on a Web page, they are automatically down-loaded due to the HTML tags. Since the main intent of Web Usage Mining is to get a picture of the user's behavior, it does not make sense to include file requests that the user did not explicitly request. Elimination of the items deemed irrelevant can be reasonably accomplished by checking the suffix oft he URL name. For instance, all log entries with filename suffixes such as, gif, jpeg, GIF, JPEG, jpg, JPG, and map can be removed. In addition, common scripts such as "count.cgi" can also be removed.

4.3.2 User Identification

Next, unique users must be identified. this task is greatly complicated by the existence of local caches, corporate firewalls, and proxy servers. The Web Usage Mining methods that rely on user cooperation are the easiest ways to deal with this problem. Following are the heuristics proposed for identification of user sessions in log/site based methods:

- Each different IP address identifies a different user.
- For same IP addresses, agent log shows a change in browser software or OS. Different agent type for an IP address represents a different user.
- Use of *access logs* and *referrer logs* can be done to construct browsing paths for each user. If a page is requested that is not directly reachable by a hyperlink from any of the pages visited by the user, heuristics assume another user with same IP address.

4.3.3 Session Identification

For logs that span long periods of time, it is very likely that users will visit the Web site more than once. The goal of session identification is to divide the page accesses of each user into individual sessions.

One of the methods proposed for this is through a *timeout,* where if the time between page requests exceeds a certain limit, it is assumed that the user is starting a new session. Many commercial products use 30 minutes as the default timeout. A *timeout* of 25.5 minutes is based on empirical data.

4.3.4 Path Completion

Another problem in reliably identifying unique user sessions is determining if there are important accesses that are not recorded in the access log. This problem is referred to as *path completion*. Methods similar to those used for user identification can be used for path completion. If a page request is made that is not directly linked to the last page a user requested, the referrer log can be checked to see what page the request came from. If the page is in the user's recent request history, the assumption is that the user backtracked with the "back" button available on most browsers, calling up cached versions of the pages until a new page was requested. If the referrer log is not clear, the site topology can be used to the same effect. If more than one page in the user's history contains a link to the requested page, it is assumed that the page closest to the previously requested page is the source of the new request. Missing page references that are inferred through this method are added to the user session file. An algorithm is then required to estimate the time of each added page reference. A simple method of picking a time-stamp is to assume that any visit to a page already seen will be effectively treated as an auxiliary page. The average reference length for auxiliary pages for the site can be used to estimate the access time for the missing pages.

4.3.5 Formatting

Once the appropriate preprocessing steps have been applied to the server log, a final preparation module can be used to properly format the sessions or transactions for the type of data mining to be accomplished. For example, since temporal information is not needed for the mining of association rules, a final association rule preparation module would strip out the time for each reference, and do any other formatting of the data necessary for the specific data mining algorithm to be used.

Figure 4.2 shows a sample web server log and Figure 4.3 shows the user sessions obtained by applying the preprocessing steps applied to this log.

#	IP Address	Userid	Time	Method/ URL/ Protocol	Status	Size	Referred	Agent
1	123.456.78.9	-	[25/Apr/1998:03:04:41 -0500]	"GET A.html HTTP/1.0"	200	3290	-	Mozilla/3.04 (Win95, I)
2	123.456.78.9	-	[25/Apr/1998:03:05:34 -0500]	"GET B.html HTTP/1.0"	200	2050	A.html	Mozilla/3.04 (Win95, I)
3	123.456.78.9	-	[25/Apr/1998:03:05:39 -0500]	"GET L.html HTTP/1.0"	200	4136	-	Mozilla/3.04 (Win95, I)
4	123.456.78.9	-	[25/Apr/1998:03:06:02 -0500]	"GET F.html HTTP/1.0"	200	5096	B.html	Mozilla/3.04 (Win95, I)
5	123.456.78.9	-	[25/Apr/1998:03:06:58 -0500]	"GET A.html HTTP/1.0"	200	3290	-	Mozilla/3.01 (X11, I, IRIX6.2, IP22)
6	123.456.78.9	-	[25/Apr/1998:03:07:42 -	] "GET B.html HTTP/1.0"	200	2050	A.html	0500 Mozilla/3.01 (X11, I, IRIX6.2, IP22)
7	123.456.78.9	-	[25/Apr/1998:03:07:55 -0500] 8140	"GET R.html HTTP/1.0"	200	8140	L.html	Mozilla/3.04 (Win95, I)
8	123.456.78.9	-	[25/Apr/1998:03:09:50 -0500]	"GET C.html HTTP/1.0"		1820	A.html	Mozilla/3.01 (X11, I, IRIX6.2, IP22)
9	123.456.78.9	-	[25/Apr/1998:03:10:02 -0500]	"GET O.html HTTP/1.0"		2270	F.html	Mozilla/3.04 (Win95, I)
10	123.456.78.9	-	[25/Apr/1998:03:10:45 -0500]	"GET J.html HTTP/1.0"		9430	C.html	Mozilla/3.01 (X11, I, IRIX6.2, IP22)
11	123.456.78.9	-	[25/Apr/1998:03:12:23 -0500]	"GET G.html HTTP/1.0"		7220	B.htm	Mozilla/3.04 (Win95, I)
12	123.456.78.9	-	[25/Apr/1998:05:05:22 -0500]	"GET A.html HTTP/1.0"		3290		Mozilla/3.04 (Win95, I)
13	123.456.78.9	-	[25/Apr/1998:05:06:03 -0500]	"GET D.html HTTP/1.0"	200	1680	A.html	Mozilla/3.04 (Win95, I)

Figure 4.2: Sample server log

Task	Result
Clean Log	• A-B-L-F-A-B-RC-O-J-G-A-D
User Identification	• A-B-F-O-G-A-D • A-B-C-J • L-R
Session Identification	• A-B-F-O-G • A-D • A-B-C-J • L-R
Path Completion	• A-B-F-O-F-B-G • A-D • A-B-A-C-J • L-R

Figure 4.3

References

[1] Cooley Robert, Mobasher Bamshad, and Srivastava Jaideep, “Data Preparation for Mining World Wide Web Browsing Patterns” Department of Computer Science and Engineering University of Minnesota

[2] Gupta G.K.” Introduction to Data Mining with Case Studies”, Professor of Computer Science, Monash University, Clayton, Autralia, PHI Publication, EEE Edition.

[3] Han Jiawei and Kamber Micheline, “Data Mining, Concepts and Techniques”

[4] Mitra Sushmita and Acharya Tinku, “Data Mining: Multimedia, Soft Computing and Bioinformatics”

[5] Srivastava Jaideep, “Web Mining :Accomplishments & Future Directions”

[6] Cooley Robert, Mobasher Bamshad, and Srivastava Jaideep, “Web Mining: Information and Pattern Discovery on the World Wide Web”.

[7] John R. Punin, Mukkai S. Krishnamoorthy, Mohammed J. Zaki, “Web Usage Mining - Languages and Algorithms”

[8] http://en.wikipedia.org/wiki/Apriori_algorithm

[9] http://www2.cs.uregina.ca/%7Edbd/cs831/cs831.html

.